

Package ‘DynareR’

August 11, 2020

Type Package

Title A Seamless Integration of 'R' and 'Dynare'

Version 0.1.1

Author Sagiru Mati [aut, cre]

Maintainer Sagiru Mati <smati@smati.com.ng>

Description It allows running 'Dynare' program from R Markdown. 'Dynare' is a software platform for handling a wide class of economic models, in particular dynamic stochastic general equilibrium ('DSGE') and overlapping generations ('OLG') models. This package serves as a 'Dynare' Knit-Engine for 'knitr' package. The package requires 'Dynare' 4.6.1 (<<https://www.dynare.org/>>) and 'Octave' 5.2.0 (<<https://www.gnu.org/software/octave/download.html>>). Write all your 'Dynare' commands in R Markdown chunk.

Depends R (>= 3.2.3)

Imports knitr (>= 1.20)

SystemRequirements Dynare (>= 4.6.1), Octave (=5.2.0)

Suggests rmarkdown

License GPL

URL <https://github.com/sagirumati/DynareR/>

BugReports <https://github.com/sagirumati/DynareR/issues>

Encoding UTF-8

VignetteBuilder knitr

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

LazyData true

Date/Publication 2020-08-11 16:02:14 UTC

R topics documented:

eng_dynare	2
include_IRF	3
run_dynare	4
run_model	6
run_models	7
write_dyn	8
write_mod	10

Index	12
--------------	-----------

eng_dynare	<i>DynareR: A Seamless Integration of R and Dynare</i>
------------	--

Description

This package runs on top of knitr to facilitate communication with Dynare. Run Dynare scripts from R Markdown document.

Usage

```
eng_dynare(options)
```

Arguments

options	Chunk options, as provided by knitr during chunk execution. Chunk option for this is dynare
---------	---

Details

The dynare engine can be activated via

```
knitr::knit_engines$set(dynare = DynareR::eng_dynare)
```

This will be set within an R Markdown document's setup chunk.

Value

Set of Dynare (open-source software for DSGE modelling) codes

Author(s)

Sagiru Mati, ORCID: 0000-0003-1413-3974, <https://smati.com.ng>

- Yusuf Maitama Sule (Northwest) University Kano, Nigeria
- SMATI Academy

References

- Bob Rudis (2015). Running Go language chunks in R Markdown (Rmd) files. Available at: <https://gist.github.com/hrbrmstr/9a>
- Yihui Xie (2019). knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.24.
- Yihui Xie (2015) Dynamic Documents with R and knitr. 2nd edition. Chapman and Hall/CRC. ISBN 978-1498716963
- Yihui Xie (2014) knitr: A Comprehensive Tool for Reproducible Research in R. In Victoria Stodden, Friedrich Leisch and Roger D. Peng, editors, Implementing Reproducible Computational Research. Chapman and Hall/CRC. ISBN 978-1466561595

See Also

write_mod write_dynare run_model run_dynare

Examples

```
knitr::knit_engines$set(dynare = DynareR::eng_dynare)
library(DynareR)
```

include_IRF	<i>Embed the graphs Impulse Response Function (IRF) in R Markdown document</i>
-------------	--

Description

Use this function to include Dynare IRF into the R Markdown document

Usage

```
include_IRF(model="", IRF="", path="")
```

Arguments

- | | |
|-------|---|
| model | Object or a character string representing the name of the model excluding .mod or .dyn file extension |
| IRF | A character string for the name of the Impulse Response Function as defined in the Dynare codes. |
| path | Object or a character string representing the path to the folder for the Dynare file. The current working directory is the default path. Specify the path only if the Dynare files live in different path from the current working directory. |

Value

Set of Dynare (open-source software for DSGE modelling) outputs

Author(s)

Sagiru Mati, ORCID: 0000-0003-1413-3974, <https://smati.com.ng>

- Yusuf Maitama Sule (Northwest) University Kano, Nigeria
- SMATI Academy

References

Bob Rudis (2015). Running Go language chunks in R Markdown (Rmd) files. Available at: <https://gist.github.com/hrbrmstr/9a>

Yihui Xie (2019). knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.24.

Yihui Xie (2015) Dynamic Documents with R and knitr. 2nd edition. Chapman and Hall/CRC. ISBN 978-1498716963

Yihui Xie (2014) knitr: A Comprehensive Tool for Reproducible Research in R. In Victoria Stodden, Friedrich Leisch and Roger D. Peng, editors, Implementing Reproducible Computational Research. Chapman and Hall/CRC. ISBN 978-1466561595

See Also

write_mod write_dynare run_model run_dynare

Examples

```
knitr::knit_engines$set(dynare = DynareR::eng_dynare)
library(DynareR)
```

run_dynare	<i>Create and run Dynare mod file</i>
------------	---------------------------------------

Description

Use this function to create and run Dynare mod file. Use `run_dynare(model, code)` if you want the Dynare files to live in the current working directory. Use `run_dynare(model, code, path)` if you want the Dynare files to live in the path different from the current working directory.

Usage

```
run_dynare(model, code, path)
```

Arguments

model	Object or a character string representing the name of the model excluding <code>.mod</code> or <code>.dyn</code> file extension
code	Object or a character string representing the set of Dynare codes
path	Object or a character string representing the path to the folder for the Dynare file. The current working directory is the default path. Specify the path only if the Dynare files live in different path from the current working directory.

Value

Set of Dynare (open-source software for DSGE modelling) outputs

See Also

write_mod write_dyn eng_dynare run_model

Examples

```

library(DynareR)
## Not run:
DynareCodes='var y, c, k, a, h, b;
varexo e, u;
parameters beta, rho, alpha, delta, theta, psi, tau;
alpha = 0.36;
rho   = 0.95;
tau   = 0.025;
beta  = 0.99;
delta = 0.025;
psi   = 0;
theta = 2.95;
phi   = 0.1;
model;
c*theta*h^(1+psi)=(1-alpha)*y;
k = beta*(((exp(b)*c)/(exp(b(+1))*c(+1)))
          *(exp(b(+1))*alpha*y(+1)+(1-delta)*k));
y = exp(a)*(k(-1)^alpha)*(h^(1-alpha));
k = exp(b)*(y-c)+(1-delta)*k(-1);
a = rho*a(-1)+tau*b(-1) + e;
b = tau*a(-1)+rho*b(-1) + u;
end;
initval;
y = 1.08068253095672;
c = 0.80359242014163;
h = 0.29175631001732;
k = 11.08360443260358;
a = 0;
b = 0;
e = 0;
u = 0;
end;

shocks;
var e; stderr 0.009;
var u; stderr 0.009;
var e, u = phi*0.009*0.009;
end;

stoch_simul;'
model<-"example1" # This is "example1" of the `Dynare` example files
code<-DynareCodes
run_dynare(model,code)

```

```
## End(Not run)
```

run_model	<i>Run a single existing mod or dyn file.</i>
-----------	--

Description

Use `run_model(model)` if the Dynare files live in the current working directory. Use `run_model(model, path)` if the Dynare files live in the path different from the current working directory.

Usage

```
run_model(model, path = "")
```

Arguments

model	Object or a character string representing the name of the model excluding .mod or .dyn file extension
path	Object or a character string representing the path to the folder for the Dynare file. The current working directory is the default path. Specify the path only if the Dynare files live in different path from the current working directory.

Value

Set of Dynare (open-source software for DSGE modelling) outputs

See Also

`write_mod` `write_dynare` `eng_dynare` `run_dynare`

Examples

```
library(DynareR)
## Not run:
DynareCodes='var y, c, k, a, h, b;
varexo e, u;
parameters beta, rho, alpha, delta, theta, psi, tau;
alpha = 0.36;
rho   = 0.95;
tau   = 0.025;
beta  = 0.99;
delta = 0.025;
psi   = 0;
theta = 2.95;
phi   = 0.1;
model;
c*theta*h^(1+psi)=(1-alpha)*y;
k = beta*(((exp(b)*c)/(exp(b+1))*c(+1)))
```

```

        *(exp(b(+1))*alpha*y(+1)+(1-delta)*k));
y = exp(a)*(k(-1)^alpha)*(h^(1-alpha));
k = exp(b)*(y-c)+(1-delta)*k(-1);
a = rho*a(-1)+tau*b(-1) + e;
b = tau*a(-1)+rho*b(-1) + u;
end;
initval;
y = 1.08068253095672;
c = 0.80359242014163;
h = 0.29175631001732;
k = 11.08360443260358;
a = 0;
b = 0;
e = 0;
u = 0;
end;

shocks;
var e; stderr 0.009;
var u; stderr 0.009;
var e, u = phi*0.009*0.009;
end;

stoch_simul;'
model<-"example1" # This is "example1" of the `Dynare` example files
code<-DynareCodes
write_mod(model,code)
run_model(model)

## End(Not run)

```

run_models

*Run multiple **existing** mod or dyn files.*

Description

Use this function to execute multiple **existing** Dynare files. Use `run_models(file)` if the Dynare files live in the current working directory. Use `run_models(file,path)` if the Dynare files live in the path different from the current working directory.

Usage

```
run_models(model, path = "")
```

Arguments

`model` Object or a vector of character strings representing the names of the Dynare model files excluding `.mod` or `.dyn` file extension

path Object or a character string representing the path to the folder for the Dynare file. The current working directory is the default path. Specify the path only if the Dynare files live in different path from the current working directory.

Value

Set of Dynare (open-source software for DSGE modelling) outputs

See Also

write_mod write_dynare eng_dynare run_dynare

Examples

```
library(DynareR)

# Provide the list of the `Dynare` files in a vector
model=c("example1","example2","agtrend","bkk")
## Not run:

run_models(model)

## End(Not run)
```

write_dyn

write a new dyn file.

Description

Use `write_dyn(model,code)` if you want the Dynare file to live in the current working directory.
 Use `write_dyn(model,code,path)` if you want the Dynare file to live in the path different from the current working directory.

Usage

```
write_dyn(model, code, path = "")
```

Arguments

model Object or a character string representing the name of the model excluding .mod or .dyn file extension

code Object or a character string representing the set of Dynare codes

path Object or a character string representing the path to the folder for the Dynare file. The current working directory is the default path. Specify the path only if the Dynare files live in different path from the current working directory.

Value

Set of Dynare (open-source software for DSGE modelling) outputs

See Also

write_mod eng_dynare run_model run_dynare

Examples

```

library(DynareR)
## Not run:
DynareCodes='var y, c, k, a, h, b;
varexo e, u;
parameters beta, rho, alpha, delta, theta, psi, tau;
alpha = 0.36;
rho = 0.95;
tau = 0.025;
beta = 0.99;
delta = 0.025;
psi = 0;
theta = 2.95;
phi = 0.1;
model;
c*theta*h^(1+psi)=(1-alpha)*y;
k = beta*(((exp(b)*c)/(exp(b(+1))*c(+1))))
      *(exp(b(+1))*alpha*y(+1)+(1-delta)*k);
y = exp(a)*(k(-1)^alpha)*(h^(1-alpha));
k = exp(b)*(y-c)+(1-delta)*k(-1);
a = rho*a(-1)+tau*b(-1) + e;
b = tau*a(-1)+rho*b(-1) + u;
end;
initval;
y = 1.08068253095672;
c = 0.80359242014163;
h = 0.29175631001732;
k = 11.08360443260358;
a = 0;
b = 0;
e = 0;
u = 0;
end;

shocks;
var e; stderr 0.009;
var u; stderr 0.009;
var e, u = phi*0.009*0.009;
end;

stoch_simul;'
model<-"example1" # This is "example1" of the `Dynare` example files
code<-DynareCodes
write_dyn(model,code)

## End(Not run)

```

write_mod	<i>Write a new mod file.</i>
-----------	------------------------------

Description

Use `write_mod(model, code)` if you want the Dynare file to live in the current working directory. Use `write_mod(model, code, path)` if you want the Dynare file to live in the path different from the current working directory.

Usage

```
write_mod(model, code, path = "")
```

Arguments

model	Object or a character string representing the name of the model excluding <code>.mod</code> or <code>.dyn</code> file extension
code	Object or a character string representing the set of Dynare codes
path	Object or a character string representing the path to the folder for the Dynare file. The current working directory is the default path. Specify the path only if the Dynare files live in different path from the current working directory.

Value

Set of Dynare (open-source software for DSGE modelling) outputs

See Also

`write_dynare` `eng_dynare` `run_model` `run_dynare`

Examples

```
library(DynareR)
## Not run:
DynareCodes='var y, c, k, a, h, b;
varexo e, u;
parameters beta, rho, alpha, delta, theta, psi, tau;
alpha = 0.36;
rho   = 0.95;
tau   = 0.025;
beta  = 0.99;
delta = 0.025;
psi   = 0;
theta = 2.95;
phi   = 0.1;
model;
c*theta*h^(1+psi)=(1-alpha)*y;
k = beta*(((exp(b)*c)/(exp(b(+1))*c(+1))))
```

```
        *(exp(b(+1))*alpha*y(+1)+(1-delta)*k));
y = exp(a)*(k(-1)^alpha)*(h^(1-alpha));
k = exp(b)*(y-c)+(1-delta)*k(-1);
a = rho*a(-1)+tau*b(-1) + e;
b = tau*a(-1)+rho*b(-1) + u;
end;
initval;
y = 1.08068253095672;
c = 0.80359242014163;
h = 0.29175631001732;
k = 11.08360443260358;
a = 0;
b = 0;
e = 0;
u = 0;
end;

shocks;
var e; stderr 0.009;
var u; stderr 0.009;
var e, u = phi*0.009*0.009;
end;

stoch_simul;'
model<-"example1" # This is "example1" of the `Dynare` example files
code<-DynareCodes
write_mod(model,code)

## End(Not run)
```

Index

* **documentation**

- eng_dynare, [2](#)
- include_IRF, [3](#)
- run_dynare, [4](#)
- run_model, [6](#)
- run_models, [7](#)
- write_dyn, [8](#)
- write_mod, [10](#)

eng_dynare, [2](#)

include_IRF, [3](#)

run_dynare, [4](#)

run_model, [6](#)

run_models, [7](#)

write_dyn, [8](#)

write_mod, [10](#)